基于大批量训练和正交正则化的跨模态哈希方法 *

张学旺[1,2†]，周　印[1]
(1. 重庆邮电大学 软件工程学院，重庆 400065; 2. 重庆大学 微电子与通信工程学院，重庆 400044)

摘　要：基于深度学习的跨模态哈希方法都使用小批量训练方式来训练模型，然而小批量方式在每次更新参数时获取样本数量有限，不能得到很好的梯度，影响最终训练的模型的检索性能。针对此问题，提出了一个新的跨模态哈希方法，该方法使用大批量方式进行训练，并引入正交正则化来增加大批量训练的稳定性，同时考虑了哈希码的离散性，将哈希码与特征之间的距离加入到目标函数中，使得哈希码能够更加真实的表示数据。在两个广泛使用的跨模态检索数据集上的实验表明该方法比现有的几种哈希方法具有更好的性能。
关键词：跨模态哈希；大批量训练；正交正则化；哈希码和特征之间的距离
中图分类号：TP391.3　　doi: 10.19734/j.issn.1001-3695.2020.02.0040

Cross-modal hashing method based on large batch training and orthogonal regularization

Zhang Xuewang[1,2†], Zhou Yin[1]
(1. *School of Software Engineering*, *Chongqing University of Posts & Telecommunications*, *Chongqing* 400065, *China*; 2. *School of Microelectronics & Communication Engineering*, *Chongqing University*, *Chongqing* 400044, *China*)

Abstract: The cross-modal hashing methods based on deep learning use the small batch training method to train their model. However, it cannot get a good gradient using this training method due to the limited number of samples in each parameter update, which affects the retrieval performance of the final trained model. To solve the problem, this paper proposed a new cross-modal hashing, which used large batch training and introduced orthogonal regularization to increase the stability of this kind of training. And to consider the discreteness of hash codes, the objective function added the distance between hash codes and features which made hash codes to represent data more realistically. Extensive experiments on two widely used public datasets in cross-modal hashing show that this method achieves better performance than several existing hashing methods.
Key words: cross-modal hashing; large batch training; orthogonal regularization; the distance between hash codes and features

0　引言

随着互联网和多媒体技术的快速发展，产生了大量不同模态的多媒体数据，比如图像、文本、视频等。不同模态的数据可以用于描述同一个事物，多视角地展现信息，可以帮助用户获得该事物的综合理解。随着不同模态的多媒体数据快速增长，跨模态检索成为了研究热点。跨模态检索的关键在于对不同模态的多媒体数据的关系进行建模，难点主要是不同模态的多媒体数据存在异构性鸿沟，无法进行直接比较[1]。跨模态哈希方法可以有效地为不同模态的数据建立比较关系，将不同模态的数据映射到共同的汉明空间中，每个数据都被转换成一个固定长度的二进制哈希码，通过将哈希码按位异或运算，可以得到数据间的汉明距离，进而得到数据间的相似性。

随着深度学习的发展，越来越多的基于深度学习的跨模态哈希方法被提出。Jiang 等人提出深度跨模态哈希(deep cross-modal hashing, DCMH)[2]，使用深度神经网络的端到端学习框架进行特征学习，直接学习离散的二进制码。Zhang 等人提出无监督生成对抗跨模态哈希(unsupervised generative adversarial cross-modal hashing, UGACH)[3]，利用生成对抗网络(generative adversarial nets, GAN)的无监督表示学习能力，学习跨模态数据的底层流形结构，通过生成模型和判别模型的对抗训练来提高判别模型的判别能力。Deng 等人提出基于三元组深度哈希(triplet-based deep hashing, TDH)[4]，采用三元组(查询样本，正样本，负样本)的形式输入训练数据，三元组形式能够更加灵活地捕捉所有可能的语义相似性，并且还考虑了模态间和模态内的相似性。Zhang 等人提出半监督生成对抗跨模态哈希(semi-supervised cross-modal hashing by generative adversarial network ，SCH-GAN)[5]，利用生成对抗网络 GAN 进行对抗训练。

大多基于深度学习的跨模态哈希方法都采用小批量训练方式训练模型，比如文献[3, 5]训练批量大小为 64，文献[4]训练批量大小为 128，这会使得在训练过程中每次更新参数时获取样本数量有限，不能得到很好的梯度，影响最终训练的模型的检索性能。

大批量训练在每次更新参数时是基于更多的样本，会得到更好的梯度，同时也使得每轮训练时间更少。基于以上特点，越来越多学者在不同研究领域探索大批量训练。Goyal 等人[6]在 ImageNet 数据集上采用大批量方式训练 ResNet-50 网络模型，批量大小为 8192，使训练时间缩小到一小时，训练的模型精度可以和小批量训练方式训练的模型精度媲美。You 等人[7]将大批量训练扩展到了自然语言处理领域，收到了很好的效果。Brock 等人[8]将大批量训练应用到图像生成领域，批量大小设为 2048，也获得了很好的性能。但是，仅仅增加批量大小会导致训练极其不稳定[9]，还容易导致网络的泛化性能下降，出现“Generalization Gap”问题[10]。

收稿日期：2020-02-16；**修回日期：**2020-04-05　**基金项目：**国家自然科学基金资助项目(61571072)；重庆市基础研究与前沿探索专项重点项目(cstc2019jcyj-zdxmX0008)；重庆市重点产业共性关键技术创新专项重大主题专项项目(cstc2017zdcy-zdzxX0013)

作者简介：张学旺(1974-)，男(通信作者)，湖南祁东人，副教授，硕导，博士研究生，主要研究方向为大数据与智能数据处理、区块链与物联网、数据安全与隐私保护、网络通信软件(zhangxw@cqupt.edu.cn)；周印(1993-)，男，硕士研究生，主要研究方向为跨媒体检索、大数据与智能数据处理.

正交正则化能够保持矩阵的范数不变，使梯度忠实传播，防止梯度消失[11]。在图像生成研究领域，Brock 等人[11]引入正交正则化，提高了网络泛化能力；在文献[8]中，Brock 等人引入正交正则化的变体，使得截断更平滑，实现了更好的性能。在多模态检索领域，Wang 等人[12]引入正交正则化，采用小批量训练模型解决了生成哈希码的冗余问题。

由于哈希码是离散的，大部分跨模态哈希方法将哈希码的离散学习问题松弛为连续学习问题。但是数据的连续实值特征在转换为哈希码的过程中，会出现信息损失，这使得哈希码不能够很好地表示数据，影响检索性能[2]。DCMH[2]采用小批量训练模型，将哈希码加入到目标函数中，直接学习哈希码，而不进行松弛。

综合上述应用于不同领域的研究方法，本文提出了基于大批量训练和正交正则化的跨模态哈希方法。该方法包含三个主要特征：a) 模型采用三元组的方式输入数据，使用大批量训练方式进行训练，以获得更好的梯度；b) 引入正交正则化用于增加大批量训练模型的稳定性；c) 在目标函数中加入哈希码与数据特征之间的距离，通过优化目标函数，数据特征在转换成哈希码时所产生的信息损失将减小，使得哈希码能够更加真实的表示数据。

1　基于大批量训练和正交正则化的跨模态哈希

在本节中将详细描述本文所提出的方法。为了阐述更简洁，本文只考虑图像和文本两种模态的数据，其他的模态数据可以进行相应扩展。简化后有两种检索任务：a) 文本检索图像；b) 图像检索文本。假定有 k 个训练数据，$I=\{I_i\}_{i=1}^k$ 表示图像数据，$T=\{T_i\}_{i=1}^k$ 表示文本数据。$F=\{F_{I_i},F_{T_i}\}_{i=1}^k$ 为数据的低维特征，$q=\{q_{I_i},q_{T_i}\}_{i=1}^k$ 表示查询数据，$H=\{H_{I_i},H_{T_i}\}_{i=1}^k$ 为数据的哈希码，而 $\|\bullet\|_{Fro}$ 表示矩阵的 Frobenius 范数。

1.1　模型结构

本模型分为图像网络和文本网络两部分。对于图像部分，很多基于深度学习的跨模态哈希方法使用卷积神经网络(CNN)来提取图像的特征以此作为模型的图像输入值，比如文献[3, 5]，使用 VGG-19 来提取图像特征。本文也将使用 VGG-19 来提取图像特征作为图像的输入值，然后经过两层全连接层得到哈希码。而对于文本，则使用词袋模型(BoW)将其表示成向量，作为文本网络的输入值，同样经过两层全连接层得到哈希码。整个模型的框架结构如图 1 所示。

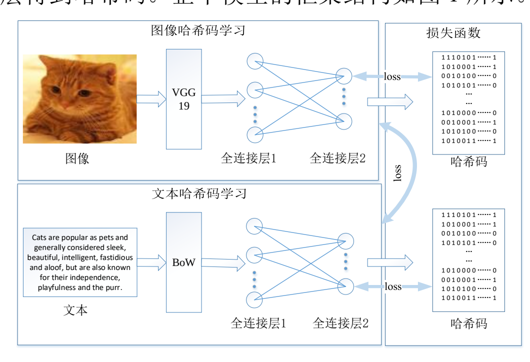


图 1　模型框架图

Fig. 1　The framework of proposed model

1.2　特征学习部分

如图 1 所示，首先将图像或者文本的特征经过第一层全连接层映射到一个共同空间，然后再经过第二层全连接层得到数据的低维特征。第一层全连接层的激活函数是 tanh 函数，而第二层全连接层的激活函数是 sigmoid 函数。整个过程可以表示如下：

$$F=sigmoid(W_{c_2}(\tanh(W_{c_1}f+B_{c_1})+B_{c_2}))\tag{1}$$

其中 W 为权重，B 为偏置项，c_1 表示第一层全连接层，c_2 表示第二层全连接层。f 表示图像的 VGG-19 特征或者文本的 BoW 向量。图像的低维特征 F_I 和文本的低维特征 F_T 维度相同，便于度量它们之间的相似性。将低维特征 F 通过阈值函数映射为哈希码 H，阈值函数如下：

$$H=\begin{cases}1, & if\ F\geq 0.5\\ 0, & if\ F<0.5\end{cases}\tag{2}$$

1.3　目标函数

由于图像检索文本任务和文本检索图像任务是对称的，所以在接下来部分仅介绍文本检索图像任务的目标函数。目标函数主要分成三部分：a) 文本特征 F_{q_T} 和图像特征 $F_I=\{F_{I^+},F_{I^-}\}$ 之间的距离；b) 哈希码 H 和特征 F 之间的距离；c) W 和 B 的正则化项。

a) 文本特征 F_{q_T} 和图像特征 $F_I=\{F_{I^+},F_{I^-}\}$ 之间的距离：

$$D_{q_{T_i}I_i^+}=\left\|F_{q_{T_i}}-F_{I_i^+}\right\|_2^2\tag{3}$$

$$D_{q_{T_i}I_i^-}=\left\|F_{q_{T_i}}-F_{I_i^-}\right\|_2^2\tag{4}$$

其中 I_i^+ 和 I_i^- 分别表示与查询文本 q_{T_i} 同语义和不同语义的图像。$D_{q_{T_i}I_i^+}$ 表示 I_i^+ 与 q_{T_i} 之间的距离。$D_{q_{T_i}I_i^-}$ 表示 I_i^- 与 q_{T_i} 之间的距离。本文使用一个基于边界的合页损失函数(a margin-based hinge loss function)来度量，如下式所示。

$$L_1=\frac{1}{n}\sum_i^n\max(0,\beta+D_{q_{T_i}I_i^+}-D_{q_{T_i}I_i^-})\tag{5}$$

其中 β 是 $D_{q_{T_i}I_i^+}$ 和 $D_{q_{T_i}I_i^-}$ 的边界值，是一个可调节的超参数。n 表示三元组 (q_{T_i},I_i^+,I_i^-) 个数。当减少损失函数 L_1 时，$D_{q_{T_i}I_i^+}$ 将会减小，而 $D_{q_{T_i}I_i^-}$ 会增大。这符合语义相似数据之间距离小，语义不同数据之间距离大的原则。在训练优化过程中，希望降低 $D_{q_{T_i}I_i^+}$ 的值，增大 $D_{q_{T_i}I_i^-}$ 的值，即 $D_{q_{T_i}I_i^+}$ 越小，$D_{q_{T_i}I_i^-}$ 越大越好。因此，可以将该过程转换为二分类问题，可以使用 sigmoid 交叉熵函数来作为损失函数。二分类问题的 sigmoid 交叉熵公式如下：

$$\begin{gathered}loss=-[z\ln(sigmoid(x)+(1-z)\ln(1-sigmoid(x)))]\\ s.t.\quad z\in\{0,1\}\end{gathered}\tag{6}$$

其中 x 表示输入值，在这里为 $D_{q_{T_i}I_i^+}$ 或者 $D_{q_{T_i}I_i^-}$。z 表示目标值。对于 $D_{q_{T_i}I_i^+}$，希望 $D_{q_{T_i}I_i^+}$ 尽可能的小，即让 $z=0$，将其带入式(6)，可以得到如下公式：

$$loss_1=-\ln(1-sigmoid(D_{q_{T_i}I_i^+}))=\ln(1+e^{D_{q_{T_i}I_i^+}})\tag{7}$$

对于 $D_{q_{T_i}I_i^-}$，希望 $D_{q_{T_i}I_i^-}$ 尽可能的大，即让 $z=1$，将其带入式(6)，可以得到如下公式：

$$loss_2=-\ln(sigmoid(D_{q_{T_i}I_i^-}))=\ln(1+e^{-D_{q_{T_i}I_i^-}})\tag{8}$$

将式(7)和(8)结合，就得到了第二个损失项：

$$L_2=\frac{1}{n}\sum_i^n(loss_1+loss_2)\tag{9}$$

损失项 L_2 和损失项 L_1 具有相似的效果，将两者结合可以更好地度量图像和文本间的相似性。

b) 哈希码 H 和特征 F 之间的距离

哈希码是离散的，当数据的实值特征 F 在被转换为哈希码 H 时，会发生信息损失：

$$D_{H_{q_{T_i}}F_{q_{T_i}}}=\left\|H_{q_{T_i}}-F_{q_{T_i}}\right\|_1\tag{10}$$

$$D_{H_{I_i}F_{I_i}}=\left\|H_{I_i}-F_{I_i}\right\|=\left\|H_{I_i^+}-F_{I_i^+}\right\|_1+\left\|H_{I_i^-}-F_{I_i^-}\right\|_1\tag{11}$$

其中 $D_{H_{q_{T_i}}F_{q_{T_i}}}$ 表示查询文本 q_{T_i} 的低维特征 $F_{q_{T_i}}$ 与其对应的哈希码 $H_{q_{T_i}}$ 之间的距离。$D_{H_{I_i}F_{I_i}}$ 表示图像 $I_i=\{I_i^+,I_i^-\}$ 的低维特征 $F_{I_i}=\{F_{I_i^+},F_{I_i^-}\}$ 与其对应的哈希码 $H_{I_i}=\{H_{I_i^+},H_{I_i^-}\}$ 之间的距离。可以得到以下损失项：

$$L_3=\frac{1}{n}\sum_i^n(D_{H_{q_{T_i}}F_{q_{T_i}}}+D_{H_{I_i}F_{I_i}})\tag{12}$$

对该损失项进行优化过程中，会使得哈希码与数据的特征越来越接近，将会减少数据特征向哈希码转换过程中的信息损失，使得哈希码更好的表达数据特征。

c) W 和 B 的正则化项

大批量训练在训练模型时不稳定，为了降低其负面影响，本文引入了正交正则化来作为权重 W 的惩罚项。对于偏置项 B，仍然使用 L2 正则化项作为惩罚项，可以得到损失项如下：

$$\begin{aligned} L_4 &= \theta\|W^{tra}W - I_{ide}\|_{Fro}^2 + \omega\|B\|_{Fro}^2 \\ &= \theta(\|W_T^{tra}W_T - I_{ide}\|_{Fro}^2 + \|W_I^{tra}W_I - I_{ide}\|_{Fro}^2) \\ &+\omega(\|B_T\|_{Fro}^2 + \|B_I\|_{Fro}^2) \end{aligned} \tag{13}$$

其中 W^{tra} 是权重矩阵 W 的转置，I_{ide} 表示单位矩阵，B 表示偏置项。W_T 表示文本网络的权重，B_T 表示文本网络的偏置项，W_I 表示图像网络的权重，B_I 表示图像网络的偏置项。而 θ 和 ω 是超参数。

将 L_1 , L_2 , L_3 和 L_4 结合在一起，就可以得到总目标函数：

$$\begin{aligned} &\min L = L_1 + \lambda L_2 + \gamma L_3 + L_4 \\ &= \frac{1}{n}\sum_i^n(\ \max(0, \beta + \|F_{q_{T_i}} - F_{I_i^+}\|_2^2 - \|F_{q_{T_i}} - F_{I_i^-}\|_2^2) + \\ &\lambda(\ln(1+e^{\|F_{q_{T_i}} - F_{I_i^+}\|_2^2}) + \ln(1+e^{-\|F_{q_{T_i}} - F_{I_i^-}\|_2^2})) + \\ &\gamma(\|H_{q_{T_i}} - F_{q_{T_i}}\|_1 + \|H_{I_i^+} - F_{I_i^+}\|_1 + \|H_{I_i^-} - F_{I_i^-}\|_1)\ ) + \\ &\theta(\|W_T^{tra}W_T - I_{ide}\|_{Fro}^2 + \|W_I^{tra}W_I - I_{ide}\|_{Fro}^2) \\ &+\omega(\|B_T\|_{Fro}^2 + \|B_I\|_{Fro}^2) \end{aligned} \tag{14}$$

其中 λ 和 γ 是超参数。

同理，图像检索文本任务的目标函数为

$$\begin{aligned} &\min L = \frac{1}{n}\sum_i^n(\ \max(0, \beta + \|F_{q_{I_i}} - F_{T_i^+}\|_2^2 - \|F_{q_{I_i}} - F_{T_i^-}\|_2^2) + \\ &\lambda(\ln(1+e^{\|F_{q_{I_i}} - F_{T_i^+}\|_2^2}) + \ln(1+e^{-\|F_{q_{I_i}} - F_{T_i^-}\|_2^2})) + \\ &\gamma(\|H_{q_{I_i}} - F_{q_{I_i}}\|_1 + \|H_{T_i^+} - F_{T_i^+}\|_1 + \|H_{T_i^-} - F_{T_i^-}\|_1)\ ) + \\ &\theta(\|W_T^{tra}W_T - I_{ide}\|_{Fro}^2 + \|W_I^{tra}W_I - I_{ide}\|_{Fro}^2) \\ &+\omega(\|B_T\|_{Fro}^2 + \|B_I\|_{Fro}^2) \end{aligned} \tag{15}$$

其中 T_i^+ 和 T_i^- 分别表示与查询图像 q_{I_i} 同语义和不同语义的文本。

1.4　优化过程

由于有图像检索文本和文本检索图像两种检索任务，因此将分别对模型进行训练，具体如下过程：

a) 首先初始化权重 W 和偏置项 B，设定批量大小为 n 和训练轮次为 e；

b) 为每个查询文本 q_{T_i} 随机取出 m 个同语义的图像 I_i^+ 和不同语义的图像 I_i^- 组成三元组 (q_{T_i}, I_i^+, I_i^-)，作为训练数据，对文本网络进行训练。固定 W_I 和 B_I，通过式(14)分别对 W_T 和 B_T 求偏导。对 W_T 求偏导为

$$\begin{aligned} &\frac{\partial \min L}{\partial W_T} = \frac{1}{n}\sum_i^n\left(\frac{\partial(\|F_{q_{T_i}} - F_{I_i^+}\|_2^2 - \|F_{q_{T_i}} - F_{I_i^-}\|_2^2)}{\partial W_T}g + \right. \\ &\lambda\frac{\partial(\ln(1+e^{\|F_{q_{T_i}} - F_{I_i^+}\|_2^2}) + \ln(1+e^{-\|F_{q_{T_i}} - F_{I_i^-}\|_2^2}))}{\partial W_T} + \\ &\left.\gamma\frac{\partial(\|H_{q_{T_i}} - F_{q_{T_i}}\|_1 + \|H_{I_i^+} - F_{I_i^+}\|_1 + \|H_{I_i^-} - F_{I_i^-}\|_1)}{\partial W_T}\right) + \\ &\frac{\partial\theta\|W_T^{tra}W_T - I_{ide}\|_{Fro}^2}{\partial W_T} \\ &= \frac{1}{n}\sum_i^n\sum_j^h\left(\frac{\partial\sum_j^h((F_{q_{T_{ij}}} - F_{I_{ij}^+})^2 - (F_{q_{T_{ij}}} - F_{I_{ij}^-})^2)}{\partial F_{q_{T_{ij}}}}g + \right. \\ &\lambda\frac{\partial(\ln(1+e^{\sum_j^h(F_{q_{T_{ij}}} - F_{I_{ij}^+})^2}) + \ln(1+e^{-\sum_j^h(F_{q_{T_{ij}}} - F_{I_{ij}^-})^2}))}{\partial F_{q_{T_{ij}}}} + \\ &\left.\gamma\frac{\partial\sum_j^h\left|H_{q_{T_{ij}}} - F_{q_{T_{ij}}}\right|}{\partial F_{q_{T_{ij}}}}\right)\frac{\partial F_{q_{T_{ij}}}}{\partial W_T} + \frac{\partial\theta\|W_T^{tra}W_T - I_{ide}\|_{Fro}^2}{\partial W_T} \end{aligned} \tag{16}$$

其中 $\|F_{q_{T_i}} - F_{I_i^+}\|_2^2 - \|F_{q_{T_i}} - F_{I_i^-}\|_2^2 > -\beta$ 时 g 为 1，否则 g 为 0。$H_{q_{T_{ij}}}$ 只有 0，1 两种值，而 $F_{q_{T_{ij}}} \in (0,1)$，所以当 $H_{q_{T_{ij}}}$ 为 0 时，$\partial\sum_j^h\left|H_{q_{T_{ij}}} - F_{q_{T_{ij}}}\right| / \partial F_{q_{T_{ij}}} = 1$；当 $H_{q_{T_{ij}}}$ 为 1 时，其值为-1。因此将 $H_{q_{T_{ij}}}$ 为 0 的值变为-1，为 1 的值保持不变，得到一个变体 $H_{q_{T_{ij}}}^{variant}$。h 表示特征的维度，也就是哈希码的长度。对于任意矩阵 A，有 $\|A\|_{Fro}^2 = tr(A^T A) = tr(AA^T)$，因此由式(16)得到：

$$\begin{aligned} \frac{\partial \min L}{\partial W_T} &= \frac{1}{n}\sum_i^n\sum_j^h\Bigg(2(F_{I_{ij}^-} - F_{I_{ij}^+})g + 2\lambda(\frac{F_{q_{T_{ij}}} - F_{I_{ij}^+}}{1+e^{-\sum_j^h(F_{q_{T_{ij}}} - F_{I_{ij}^+})^2}} \\ &+ \frac{F_{I_{ij}^-} - F_{q_{T_{ij}}}}{1+e^{\sum_j^h(F_{q_{T_{ij}}} - F_{I_{ij}^-})^2}}) - \gamma H_{q_{T_{ij}}}^{variant}\Bigg)\frac{\partial F_{q_{T_{ij}}}}{\partial W_T} + 4\theta(W_T W_T^{tra} - I_{ide})W_T \end{aligned} \tag{17}$$

同理可以得到 B_T 的偏导数，然后采用后向传播算法更新权重 W_T 和偏置项 B_T。

c) 为每个查询图像 q_{I_i} 随机取出 m 个同语义的文本 T_i^+ 和不同语义的文本 T_i^- 组成三元组 (q_{I_i}, T_i^+, T_i^-)，作为训练数据，对图像网络进行训练。固定 W_T 和 B_T，与求 W_T 和 B_T 的偏导数类似，通过式(15)求偏导，采用后向传播算法更新权重 W_I 和偏置项 B_I。

整个方法如算法 1 所示。

算法 1　算法描述

输入：训练数据 I ，T；

输出：权重 W 和偏置项 B；

随机初始化权重 W 和 B，训练批量大小设为 b，训练轮次设为 e；

for epoch=0,1,2, …, e-1 **do**

　if epoch%30==0 **do**

　　for $q_{T_i} = T_1, T_2, \cdots, T_k$ **do**

　　　随机取出 m 个与查询文本 q_{T_i} 相关的正例图像 I_i^+ 和与查询文本 q_{T_i} 不相关的负例图像 I_i^- 组成 m 个三元组 (q_{T_i}, I_i^+, I_i^-)，作为训练数据。

　　end for

　end if

for step=1,2⋯,$\lceil k*m/b \rceil$ **do**

　固定 W_I 和 B_I，通过式(14)求偏导，采用后向传播算法更新权重 W_T 和偏置项 B_T。

　end for

　if epoch%30==0 **do**

　　for $q_{I_i} = I_1, I_2, \cdots, I_k$ **do**

　　　随机取出 m 个与查询图像 q_{I_i} 相关的正例文本 T_i^+ 和与查询图像 q_{I_i} 不相关的负例文本 T_i^- 组成 m 个三元组 (q_{I_i}, T_i^+, T_i^-)，作为训练数据。

end for

　end if

　for step=1,2⋯,$\lceil k*m/b \rceil$ **do**

　固定 W_T 和 B_T，通过式(15)求偏导，采用后向传播算法更新权重 W_I 和偏置项 B_I

　end for

end for

2　实验

在本章中，将在两个广泛用于跨模态哈希的数据集上评估本文方法的性能，并与几个比较先进的方法进行对比。

2.1　数据集

本文分别在 Wikipedia[13]和 MIRFlickr[14]数据集上开展实验。Wikipedia 数据集是一个流行的数据集，它由 10 个类别

共 2866 个文本/图像数据对组成。本文将其中 2173 个数据对作为训练数据集和检索数据集，剩余的 693 个数据对作为测试数据集。每个图像由深度学习框架 Keras 应用程序中的 19 层 VGGNet 的 fc2 层提取的 4096 维特征表示，而每个文本由 1000 维的 BoW 的向量表示。

MIRFlickr 数据集从 Flickr 网站上收集得到，包含 25000 个文本/图像数据对，分为 24 个类别。但是其为人工标注，有些图像没有文本描述，有些图像没有类别，需要对其进行筛选。首先预处理文本，移除标点符号和停用词，然后统计各个单词的次数，取出现 20 次以上的单词组成 BoW 的词汇表，移除不包含在词汇表中单词的文本/图像数据对，同时也移除没有文本描述的图像或者没有类别的文本/图像数据对。经过这些处理后还剩余 20819 个文本/图像数据对。随机取出其中的 5%的数据对作为测试数据集，剩余数据对作为检索数据集，并随机取出 5000 个数据对作为训练数据集。每个图像都由 Keras 应用程序中的 19 层 VGGNet 的 fc2 层提取的 4096 维特征表示，而每个文本由 1386 维的 BoW 的向量表示。相关统计信息见表 1。

表 1　两个基准数据集的统计信息

Tab. 1　Statistics of two benchmark datasets

	Wikipedia	MIRFlickr
数据集大小	2866	20819
训练集	2173	5000
测试集	693	1041
检索集	2173	19778
类别	10	24

2.2　评价指标

本文在每个数据集上执行两种检索任务:图像检索文本和文本检索图像，分别用 image→text 和 text→image 表示。本文使用两个广泛使用的评价标准来评价跨模态哈希的检索性能，如下所示。

a) mean average precision (MAP)：表示所有查询数据的 average precisions (AP)的均值。其中 AP 表示查询的平均准确率，定义如下：

$$AP = \frac{1}{R}\sum_{t=1}^{s}\frac{R_t}{t}\times rel_t \tag{18}$$

其中 R 表示检索数据集中所有的相关数据的数量，s 表示检索数据集中所有的数据的数量，R_t 表示检索出的前 t 个数据中的相关数据的数量，rel_t 表示第 t 个数据是否是相关数据，如果为 1，则相关，如果为 0，则不相关。

b) Precision-recall 曲线 (PR-曲线)：表示准确率召回率曲线，在不同召回率下的检索准确率，常常用于评价检索性能。

2.3　对比方法和实现细节

本文与两个非深度学习方法 SePH[15]和 GSPH[16]进行了对比，SePH 和 GSPH 都是基于核的，使用核逻辑回归(KLR)学习哈希函数来取得最好的结果，都分别采用了 k-均值算法和随机抽样法。因此，在对比这两种方法时，用 klr+k 表示使用 k-均值算法的核逻辑回归，而用 klr+r 表示使用随机抽样的核逻辑回归。另外还与现有三个基于深度学习的跨模态哈希方法 DCMH[2]，UGACH[3]和 SCH-GAN[5] 进行了对比，DCMH 采用端到端网络结构，直接学习哈希码，UGACH 和 SCH-GAN 都基于生成对抗网络 GAN，采用三元组方式输入数据。在本文实验中要用到图像和文本两种模态数据，对比时在将一种模态的数据作为查询数据集的时候，将另一种模态的数据作为检索数据集。为了公平对比，所有方法都采用 3.1 节中描述的图像和文本的表示形式作为输入值，即图像值是 4096 维特征，文本是 BoW 向量。DCMH 作为一种端到端的方法，可以直接输入图像，本文用 $DCMH_{vgg19}$ 表示前述对比形式，另外用 $DCMH_{original}$ 表示直接将图像作为输入值进行训练。对比的五个方法的代码都由相应的作者提供。对于本文的方法，参考文献[5]将超参数设置为：$\lambda=0.01$，$\gamma=0.01$ 和 $\omega=0.01$；参考文献[8]，令 $\theta=0.0001$；对于 β，哈希码位数不同，在汉明空间中的维度就不同，位数越大，维度越大，边界值越大才能更好的在汉明空间中将正例数据和负例数据分开。哈希码位数分别为 16、32 和 64 位(bits)，β 分别为 6、8 和 10。在 Wikipedia 数据集上的学习率为 0.08，在 MIRFlickr 数据集的学习率为 0.016。与文献[3, 5]类似，对于每一个数据挑选出相应的数据组成 4 个三元组用于训练，即将 m 设为 4，因此对于 Wikipedia 数据集总共有 2173*4=8692 个三元组，对于 MIRFlickr 数据集总共有 5000*4=20000 个三元组。批量大小 b 设为 8192。本文使用 tensorflow 深度学习框架来实现代码，具体的软件版本是 python3.5.2 和 tensorflow1.11.0。所有的实验都是在 NVIDIA GTX 1080Ti 图形卡，Intel(R) Xeon(R) E5-2620 v4 2.10GHz CPU, 128 GB 内存的机器上运行得到。

2.4　实验结果

表 2 和 3 分别展示了各个方法在 MIRFlickr 和 Wikipedia 数据集上的 MAP 的实验结果。从结果可以看出，对于 32 位和 64 位的哈希码，本文的方法在两个数据集上都取得了最好的结果。总体来看，与第二好的方法 SCH-GAN 相比，在图像检索文本的任务中，本文的方法在 MIRFlickr 和 Wikipedia 数据集上分别比其大约高出了 1.8%和 8.2%；而在文本检索图像的任务中，则分别大约提高了 1.3%和 2%。这主要是因为本文使用大批量训练模型，可以得到更好的梯度，并使用正交正则化使训练更加稳定。还因为将哈希码与数据特征之间的距离添加到损失函数中，使得哈希码更真实地表示数据的特征。对于 16 位的哈希码，由于其长度不能充分表示数据特征，尽管本文的方法在 Wikipedia 数据集上取得了最好的成绩，但在 MIRFlickr 数据集上只是第二好，说明哈希码长度对 MAP 具有一定的影响。

表 2　在 MIRFlickr 数据集上的 MAP

Tab. 2　The MAP scores on mirflickr dataset

方法	image→text			text→image		
	16	32	64	16	32	64
$SePH_{klr+r}$ [15]	0.7364	0.7367	0.7451	0.7486	0.7514	0.7573
$SePH_{klr+k}$ [15]	0.7377	0.7459	0.7467	0.7522	0.7595	0.7599
$GSPH_{klr+r}$ [16]	0.7279	0.7425	0.7541	0.7579	0.7693	0.7760
$GSPH_{klr+k}$ [16]	0.7374	0.7485	0.7584	0.7614	0.7729	0.7798
UGACH [3]	0.6100	0.6045	0.5848	0.6278	0.6029	0.6101
$DCMH_{original}$ [2]	0.7296	0.7363	0.7386	0.7639	0.7650	0.7703
$DCMH_{vgg19}$ [2]	**0.7433**	0.7527	0.7592	0.7669	0.7792	0.7837
SCH-GAN [5]	0.7203	0.7481	0.7609	0.7661	0.7851	0.7884
本文算法	0.7410	**0.7571**	**0.7718**	**0.7822**	**0.7915**	**0.7953**

表 3　在 Wikipedia 数据集上的 MAP

Tab. 3　The MAP scores on Wikipedia dataset

方法	image→text			text→image		
	16	32	64	16	32	64
$SePH_{klr+r}$ [15]	0.5009	0.5287	0.5393	0.5508	0.5955	0.6190
$SePH_{klr+k}$ [15]	0.4997	0.5252	0.5413	0.5584	0.6009	0.6122
$GSPH_{klr+r}$ [16]	0.5064	0.5289	0.5320	0.5701	0.6001	0.6237
$GSPH_{klr+k}$ [16]	0.5117	0.5318	0.5390	0.5801	0.6036	0.6207
UGACH [3]	0.3332	0.3605	0.3688	0.3222	0.3323	0.3471
$DCMH_{original}$ [2]	0.4503	0.4506	0.4120	0.7419	0.7238	0.6940
$DCMH_{vgg19}$ [2]	0.4387	0.4698	0.4809	0.8279	0.8457	0.7927
SCH-GAN [5]	0.5207	0.5370	0.5076	0.8352	0.8351	0.8288
本文算法	**0.5528**	**0.5712**	**0.5688**	**0.8426**	**0.8502**	**0.8572**

图 2 和 3 分别给出了各个方法在 Wikipedia 和 MIRFlickr 数据集上的 32 位和 64 位哈希码所对应的 PR-曲线。从图中可以看出，本文的方法比其他方法都要好。

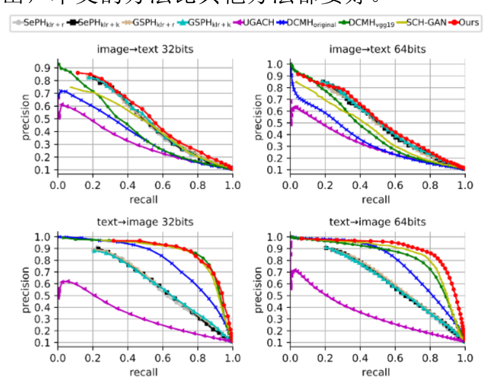


图 2　在 Wikipedia 数据集上的 PR 曲线图

Fig. 2　The PR-curves on Wikipedia dataset

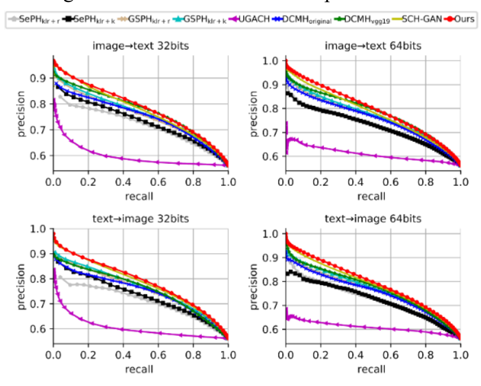


图 3　在 MIRFlickr 数据集上的 PR 曲线图

Fig. 3　The PR-curves on mirflickr dataset

此外，本文还在 Wikipedia 数据集上比较了不同批量大小和正交正则化对模型训练的影响。批量大小分别设置为 512，2048 和 8192。提高批量大小需要增加学习率，才能保证收敛速度，因此学习率分别为 0.02、0.04 和 0.08。N 表示不使用正交正则化，Y 表示使用正交正则化。当不使用正交正则化时，用 L2 正则化来代替。即式(13)被替换为

$$L_4 = \theta(\|W_T\|_{Fro}^2 + \|W_I\|_{Fro}^2) + \omega(\|B_T\|_{Fro}^2 + \|B_I\|_{Fro}^2) \tag{19}$$

其余参数配置都一样，所有的实验结果如表 4 所示。在评价性能好坏时，不仅需要看图像检索文本任务的 MAP，也需要考虑文本检索图像任务的 MAP，因此表 4 也给出了两个任务总的 MAP。从表中可以看出，仅仅增大批量效果并不好，甚至还有所降低，这主要因为批量增大后，模型训练不稳定，网络泛化能力下降；而在使用同一个批量大小进行训练时，如果使用了正交正则化，会得到更好的果，但是每轮训练时间会增加。而增大批量可以加快训练速度，降低每轮训练的时间。总体上来看，当批量大小为 8192，使用了正交正则化，本文方法得到了最好的结果。

3　结束语

在目前广泛应用的跨模态检索的研究领域，本文提出了一个新的跨模态哈希方法——基于大批量训练和正交正则化的跨模态哈希方法。该方法采用三元组的方式输入数据，使用大批量训练方式进行训练，引入正交正则化使得模型训练更加稳定，并且还度量了哈希码和数据特征之间的距离，使得哈希码能够更好地表示数据。在两个广泛使用的 Wikipedia 和 MIRFlickr 数据集上的实验结果证明了该方法的有效性，相比现有的几种哈希方法具有更好的性能。

表 4　在 Wikipedia 数据集上的不同批量大小训练

Tab. 4　The different batch size training on Wikipedia dataset

批量大小	学习率	是否使用正交正则化	MAP (image→text)			MAP (text→image)			总 的 MAP			每轮训练时间		
			16	32	64	16	32	64	16	32	64	16	32	64
512	0.02	N	0.5406	0.5506	0.5612	0.8280	0.8524	0.8432	1.3686	1.4030	1.4044	9.3s	9.6s	10.2s
512	0.02	Y	**0.5588**	0.5629	0.5691	0.8312	**0.8576**	0.8479	1.3900	1.4205	1.4170	16.0s	16.3s	16.8s
2048	0.04	N	0.5509	0.5594	0.5699	0.8231	0.8462	0.8326	1.3740	1.4056	1.4025	8.1s	8.2s	8.5s
2048	0.04	Y	0.5559	0.5668	**0.5702**	0.8380	0.8486	0.8545	1.3939	1.4154	1.4247	10.9s	11.0s	11.3s
8192	0.08	N	0.5487	0.5563	0.5649	0.8217	0.8431	0.8327	1.3704	1.3994	1.3976	7.5s	7.5s	7.9s
8192	0.08	Y	0.5528	**0.5712**	0.5688	**0.8426**	0.8502	**0.8572**	**1.3954**	**1.4214**	**1.4260**	8.2s	8.3s	8.6s

参考文献：

[1] Peng Yuxin, Huang Xin and Zhao Yunzhen. An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges [J]. IEEE Trans on Circuits and Systems for Video Technology, 2018, 28 (9): 2372-2385.

[2] Jiang Qing Yuan and Li Wu Jun. Deep cross-modal hashing [C]// Proc of the 30th IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2017: 3232–3240.

[3] Zhang Jian, Peng Yuxin and Yuan Mingkuan. Unsupervised generative adversarial cross-modal hashing [C]// Proc of the 32nd AAAI International Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2018: 539-546.

[4] Deng Cheng, Chen Zhaojia, Liu Xianglong, *et al.* Triplet-Based Deep Hashing Network for Cross-Modal Retrieval [J]. IEEE Trans on Image Processing, 2018, 27 (8): 3893-3903.

[5] Zhang Jian, Peng Yuxin and Yuan Mingkuan. SCH-GAN: Semi-supervised cross-modal hashing by generative adversarial network [J]. IEEE Trans on Cybernetics, 2020, 50 (2): 489-502.

[6] Goyal P, Dollár P, Girshick R, *et al.* Accurate large minibatch SGD: Training imagenet in 1 hour [EB/OL]. (2018-04-30) [2020-02-16]. https://ar xiv. org/abs/1706. 02677.

[7] You Yang, Hseu J, Ying C, *et al.* Large-batch training for LSTM and beyond [C]// Proc of the 26th International Conference for High Performance Computing, Networking, Storage and Analysis. New York: ACM Press, 2019: 1-16.

[8] Brock A, Donahue J and Simonyan K. Large scale GAN training for high fidelity natural image synthesis [C/OL]/ Proc of the 7th International Conference on Learning Representations. 2019. (2019-02-26) [2020-02-16] https://openreview. net/pdf?id=B1xsqj09Fm.

[9] You Yang, Gitman I and Ginsburg B. Large batch training of convolutional networks [EB/OL]. (2017-09-13) [2020-02-16]. https://arxiv. org/abs/1708. 0 3888.

[10] Keskar N S, Mudigere D, Nocedal J, *et al.* On large-batch training for deep learning: Generalization gap and sharp minima [C/OL]/ Proc of the 5th International Conference on Learning Representations. 2017. (2017-

02-10) [2020-02-16]. https://openreview. net/pdf?id=H1oyRlYgg.

[11] Brock A, Lim T, Ritchie J M, *et al.* Neural photo editing with introspective adversarial networks [C/OL]/ Proc of the 5th International Conference on Learning Representations. 2017. (2017-02-23) [2020-02-16]. https://openre view. net/pdf?id=HkNKFiGex.

[12] Wang Daixin, Peng Cui, Ou Mingdong, *et al.* Deep multimodal hashing with orthogonal regularization [C]// Proc of the 24th AAAI International Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2015: 2291-2297.

[13] Pereira J C, Coviello E, Doyle G, *et al.* On the role of correlation and abstraction in cross-modal multimedia retrieval [J]. IEEE Trans on pattern analysis and machine intelligence, 2014, 36 (3): 521-535.

[14] Huiskes M J and Lew M S. The MIR flickr retrieval evaluation [C]// Proc of the 1st ACM International Conference on Multimedia Information Retrieval. New York: ACM Press, 2008: 39-43.

[15] Lin Zijia, Ding Guiguang, Hu Mingqing, *et al.* Semantics-Preserving Hashing for Cross-View Retrieval [C]// Proc of the 28th IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2015: 3864-3872.

[16] Mandal D, Chaudhury K N and Biswas S. Generalized semantic preserving hashing for cross-modal retrieval [J]. IEEE Trans on Image Processing, 2019, 28 (1): 102-112.